



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/777,909	02/12/2004	William Preston Alexander III	AUS920030825US1	6082
35525	7590	08/17/2011		
IBM CORP (YA) C/O YEE & ASSOCIATES PC P.O. BOX 802333 DALLAS, TX 75380			EXAMINER RAMPURIA, SATISH	
			ART UNIT 2191	PAPER NUMBER
			NOTIFICATION DATE 08/17/2011	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeciipaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte WILLIAM PRESTON ALEXANDER III, ROBERT TOD
DIMPSEY, FRANK ELLIOT LEVINE, ROBERT JOHN URQUHART

Appeal 2009-010758
Application 10/777,909
Technology Center 2100

Before JAY P. LUCAS, LANCE LEONARD BARRY, and
ST. JOHN COURTENAY III, *Administrative Patent Judges*.

COURTENAY, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

This is an appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1 and 4-8, which are all the claims pending in the application. Claims 2, 3, and 9-24 were cancelled during prosecution. We have jurisdiction under 35 U.S.C. § 6(b).

We Reverse.

Invention

Appellants' invention relates generally to an improved computing system for tracing the execution of a computer program. More particularly, the invention on appeal is directed to a method and apparatus for identifying differences in runs of a computer program due to changes in the code of the computer program. (Spec. 1).

Representative Claim 1

A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program, comprising:

obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program;

generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and

outputting the minimized call tree data structure.

Rejection

Claims 1 and 4-8 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the combination of Alexander (US 6,338,159 B1) and Kazi et al., "JaViz: A client/server Java profiling tool", IBM Systems Journal 39, No.1, 2000, Pages: 96-117.

ISSUE

Based upon our review of the administrative record, we have determined that the following issue is dispositive in this appeal:

Under §103, did the Examiner err in finding that the combination of Alexander and Kazi would have taught or suggested the following limitations of claim 1:

obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; [and]

generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures[.]

ANALYSIS

At the outset, we observe that the plain language of claim 1 requires “obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program.” We observe that “the computer program” has clear antecedent basis in the recitation of “a computer program” in the preamble. (claim 1). Thus, under a broad but reasonable interpretation, we conclude that claim 1 requires obtaining *a*

plurality of call tree data structures from the plurality of executions of the same program.¹

Based on our review of the record, we agree with Appellants that the cited combination of Alexander and Kazi does not teach or suggest obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the (same) computer program. We agree with Appellants that this is the fundamental shortcoming in the Examiner's proffered prima facie case.

In the response to arguments section of the Answer, the Examiner avers that "Alexander uses the word ['']programs[''] as [a] plural form which represents one or more programs, thus the tracing is performed on a *plurality of programs execution*." (Ans. 11, ¶ 2)(emphasis added). However, we conclude that the plain language of claim 1 requires a *plurality of call tree data structures* that are obtained from the plurality of executions of the same program, as discussed above.

While the Alexander reference does teach using a call stack tree (e.g., Fig. 4) for "profiling the performance characteristics of long running programs" (col. 6, ll. 51-52) and "long-running applications" (col. 6, l. 60), we do not find, and the Examiner has not established, that a *plurality of call tree data structures* are obtaining from the plurality of executions of the same program, as required by the language of claim 1.

As pointed out by Appellants (App. Br. 11), Alexander teaches that "an accurate and informative performance profile may be obtained for a long

¹ We accord Appellants' claims the broadest reasonable interpretation consistent with the Specification. See *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

running program.” (Alexander, col. 2, ll. 59-60). Therefore, we agree with the Appellants that “Alexander is directed to providing trace information in connection with *a single execution* of a long running program.” (App. Br. 11, ¶1)(emphasis added).

Although the secondary Kazi reference does teach the use of call trees (Fig. 2, p. 101), and also analyzing trace files to gather the total number of calls made to each method, including the maximum, minimum, and average execution times (pg. 100, 1st col., last para.), we agree with Appellants that Kazi suffers from the same deficiency discussed above regarding the primary Alexander reference.

Given that the Examiner’ s proffered combination of Alexander and Kazi does not teach nor fairly suggest obtaining *a plurality of call tree data structures* from the plurality of executions of the same program, we also agree with Appellants that the combination of Alexander and Kazi does not teach or suggest “generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes *a minimum set of nodes that are consistent between the plurality of call tree data structures*,” which is the central inventive aspect in the claims before us on appeal.

In particular, we do not agree with the Examiner that pruning or performing a reduction of a single call tree (as taught by Alexander, col 2, l. 5), or merely finding an average or minimum execution time of a node (as taught by Kazi, 100), teaches or fairly suggests generating a minimized call tree data structure from the plurality of call tree data structures wherein the minimized call tree data structure includes *a minimum set of nodes that are consistent between the plurality of call tree data structures*.

Within the meaning of Appellants' independent claim 1, we conclude that the generation of the claimed minimized call tree data structure implicitly requires a comparison between nodes (to determine consistency between nodes) of different call tree data structures that were obtained from the plurality of executions of the same program. Thus, on this record, we agree with Appellants that the Examiner's proffered combination of Alexander and Kazi falls short of rendering the claimed invention obvious.

For essentially the same reasons argued by Appellants in the Briefs, we agree that the Examiner erred in rejecting independent claim 1 and associated dependent claims 4-8. Accordingly, we reverse the Examiner's rejection of claim 1, and associated dependent claims 4-8 which stand therewith.

DECISION

We reverse the § 103 rejection of claims 1 and 4-8.

ORDER REVERSED

Tkl